Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

Claim 1 (currently amended): Instruction Set Architecture (ISA) selection logic within a CPU for selecting an ISA decoding mode for a program instruction, the selection logic comprising:

a plurality of boundary address registers for storing boundary addresses that partition an address space into a plurality of address ranges, each of the plurality of address ranges corresponding to one of a plurality of ISA decoding modes; and

an ISA mode controller, coupled to the plurality of boundary address registers, that includes address evaluation logic,

wherein the ISA mode controller

receives an address of a program instruction to be decoded,

sequentially compares the address to boundary addresses stored in the plurality of boundary address registers, and

determines an ISA decoding mode for the program instruction based upon the comparison of the address to the boundary addresses.

Claims 2-8 (cancelled)

Claim 9 (currently amended): The ISA selection logic of claim 1, wherein the ISA mode controller provides the ISA decoding mode to an instruction decoder to enable correct decoding of the program instruction.

Claims 10-17 (cancelled)

Claim 18 (currently amended): A multiple-ISA mode processor, comprising:

an ISA mode controller that includes address evaluation logic;

a plurality of boundary address registers, coupled to the ISA mode controller; and

an instruction decoder, coupled to the ISA mode controller,

wherein the ISA mode controller

receives an address of a program instruction to be decoded,

sequentially compares the address to boundary addresses stored in the plurality of boundary address registers that partition an address space of the processor into a plurality of address ranges,

determines an ISA decoding mode corresponding to one of the plurality of address ranges for the program instruction based upon the comparison of the address to the boundary addresses, and

provides the ISA decoding mode for the program instruction to the instruction decoder.

Claims 19-35 (cancelled)

Claim 36 (previously presented): The ISA selection logic of claim 1, wherein the plurality of boundary address registers store boundary addresses that represent lower address bounds for the plurality of address ranges.

Claim 37 (previously presented): The ISA selection logic of claim 1, wherein the plurality of boundary address registers store boundary addresses that represent upper address bounds for the plurality of address ranges.

Claim 38 (previously presented): The ISA selection logic of claim 1, wherein the plurality of boundary address registers store boundary addresses that partition the address space into a plurality of unequal address ranges.

Claim 39 (previously presented): The ISA selection logic of claim 1, wherein boundary addresses are written to the plurality of boundary address registers during initialization of the CPU.

Claim 40 (previously presented): The ISA selection logic of claim 1, wherein boundary addresses are altered by an operating system as application programs are fetched and loaded into memory.

Claim 41 (previously presented): The ISA selection logic of claim 1, wherein software within a linker/loader program determines the boundary addresses that are loaded into the plurality of boundary address registers.

Claim 42 (cancelled)

Claim 43 (currently amended): The processor of claim ~~20~~ 18, wherein the plurality of boundary address registers store boundary addresses that represent lower address bounds for the plurality of address ranges.

Claim 44 (currently amended): The processor of claim ~~20~~ 18, wherein the plurality of boundary address registers store boundary addresses that represent upper address bounds for the plurality of address ranges.

Claim 45 (currently amended): The processor of claim ~~20~~ 18, wherein the plurality of boundary address registers store boundary addresses that partition the address space into a plurality of unequal address ranges.

Claim 46 (currently amended): The processor of claim ~~20~~ 18, wherein boundary addresses are written to the plurality of boundary address registers during initialization of the processor.

Claim 47 (currently amended): The processor of claim ~~20~~ 18, wherein boundary addresses are altered by an operating system as application programs are fetched and loaded into memory.

Claim 48 (currently amended): The processor of claim ~~20~~ 18, wherein software within a linker/loader program determines the boundary addresses that are loaded into the plurality of boundary address registers.

Claim 49-57 (cancelled)

Claim 58 (currently amended): A method for determining ISA decoding modes for program instructions of a multiple-ISA application program running on a processor, wherein the application program includes instructions associated with a first ISA mode requiring a first amount of memory space and instructions associated with a second ISA mode requiring a second amount of memory space, and wherein the processor includes a plurality of boundary address registers and an ISA mode controller coupled to the plurality of boundary address registers, the method comprising:

(1) storing application program instructions associated with the first ISA mode in memory beginning at a first memory address;

(2) writing the first memory address to a first boundary address register of the plurality of boundary address registers, wherein the first memory address acts as a first boundary address that partitions the memory and creates a first memory address range;

(3) storing application program instructions associated with the second ISA mode in memory beginning at a second memory address;

(4) writing the second memory address to a second boundary address register of the plurality of boundary address registers, wherein the second memory address acts as a

second boundary address that partitions the memory and creates a second memory address range;

(5) retrieving a program instruction form a third memory address;

(6) sequentially comparing the third memory address with the first memory address in the first boundary address register and the second memory address in the second boundary address register to determine whether the third memory address corresponds to the first memory address range or the second memory address range;

(7) generating, if the third memory address corresponds to the first memory address range, a first ISA mode indicator output with the ISA mode controller; and

(8) generating, if the third memory address corresponds to the second memory address range, a second ISA mode indicator output with the ISA mode controller.

Claim 59 (previously presented): The method of claim 58, wherein a memory address having a value greater than the first boundary address and less than the second boundary address corresponds to the first memory address range.

Claim 60 (previously presented): The method of claim 58, wherein a memory address having a value less than the first boundary address and greater than the second boundary address corresponds to the first memory address range.

Claim 61 (previously presented): The method of claim 58, wherein step (2) and step (4) occur only during initialization of the processor.

Claim 62 (previously presented): The method of claim 58, wherein step (2) and step (4) are performed by an operating system as application programs are fetched and loaded into memory.

Claims 63-67 (cancelled)

Claim 68 (new): Instruction Set Architecture (ISA) selection logic within a CPU for selecting an ISA decoding mode for a program instruction, the selection logic comprising:

a plurality of boundary address registers for storing boundary addresses that partition an address space into a plurality of address ranges, each of the plurality of address ranges corresponding to one of a plurality of ISA decoding modes; and

an ISA mode controller, coupled to the plurality of boundary address registers, that includes address evaluation logic,

wherein the ISA mode controller

receives an address of a program instruction to be decoded,

compares, in parallel, the address to boundary addresses stored in the plurality of boundary address registers, and

determines an ISA decoding mode for the program instruction based upon the comparison of the address to the boundary addresses.

Claim 69 (new): The ISA selection logic of claim 68, wherein the ISA mode controller provides the ISA decoding mode to an instruction decoder to enable correct decoding of the program instruction.

Claim 70 (new): The ISA selection logic of claim 68, wherein the plurality of boundary address registers store boundary addresses that represent lower address bounds for the plurality of address ranges.

Claim 71 (new): The ISA selection logic of claim 68, wherein the plurality of boundary address registers store boundary addresses that represent upper address bounds for the plurality of address ranges.

Claim 72 (new): The ISA selection logic of claim 68, wherein the plurality of boundary address registers store boundary addresses that partition the address space into a plurality of unequal address ranges.

Claim 73 (new): The ISA selection logic of claim 68, wherein boundary addresses are written to the plurality of boundary address registers during initialization of the CPU.

Claim 74 (new): The ISA selection logic of claim 68, wherein boundary addresses are altered by an operating system as application programs are fetched and loaded into memory.

Claim 75 (new): The ISA selection logic of claim 68, wherein software within a linker/loader program determines the boundary addresses that are loaded into the plurality of boundary address registers.

Claim 76 (new): A multiple-ISA mode processor, comprising:

an ISA mode controller that includes address evaluation logic;

a plurality of boundary address registers, coupled to the ISA mode controller; and

an instruction decoder, coupled to the ISA mode controller,

wherein the ISA mode controller

receives an address of a program instruction to be decoded,

compares, in parallel, the address to boundary addresses stored in the plurality of boundary address registers that partition an address space of the processor into a plurality of address ranges,

determines an ISA decoding mode corresponding to one of the plurality of address ranges for the program instruction based upon the comparison of the address to the boundary addresses, and

provides the ISA decoding mode for the program instruction to the instruction decoder.

Claim 77 (new): The processor of claim 76, wherein the plurality of boundary address registers store boundary addresses that represent lower address bounds for the plurality of address ranges.

Claim 78 (new): The processor of claim 76, wherein the plurality of boundary address registers store boundary addresses that represent upper address bounds for the plurality of address ranges.

Claim 79 (new): The processor of claim 76, wherein the plurality of boundary address registers store boundary addresses that partition the address space into a plurality of unequal address ranges.

Claim 80 (new): The processor of claim 76, wherein boundary addresses are written to the plurality of boundary address registers during initialization of the processor.

Claim 81 (new): The processor of claim 76, wherein boundary addresses are altered by an operating system as application programs are fetched and loaded into memory.

Claim 82 (new): The processor of claim 76, wherein software within a linker/loader program determines the boundary addresses that are loaded into the plurality of boundary address registers.

Claim 83 (new): A method for determining ISA decoding modes for program instructions of a multiple-ISA application program running on a processor, wherein the application program includes instructions associated with a first ISA mode requiring a first amount of memory space and instructions associated with a second ISA mode requiring a second amount of memory space, and wherein the processor includes a plurality of boundary address registers and an ISA mode controller coupled to the plurality of boundary address registers, the method comprising:

(1) storing application program instructions associated with the first ISA mode in memory beginning at a first memory address;

(2) writing the first memory address to a first boundary address register of the plurality of boundary address registers, wherein the first memory address acts as a first boundary address that partitions the memory and creates a first memory address range;

(3) storing application program instructions associated with the second ISA mode in memory beginning at a second memory address;

(4) writing the second memory address to a second boundary address register of the plurality of boundary address registers, wherein the second memory address acts as a second boundary address that partitions the memory and creates a second memory address range;

(5) retrieving a program instruction form a third memory address;

(6) comparing, in parallel, the third memory address with the first memory address in the first boundary address register and the second memory address in the second boundary address register to determine whether the third memory address corresponds to the first memory address range or the second memory address range;

(7) generating, if the third memory address corresponds to the first memory address range, a first ISA mode indicator output with the ISA mode controller; and

(8) generating, if the third memory address corresponds to the second memory address range, a second ISA mode indicator output with the ISA mode controller.